self-service-machine-api - STEP 2

Nous voulons mettre en place notre 1ère route (ou point de terminaison).

Pour l'instant, nous allons simuler la base de données (Mocker) avec un fichier `mock-product.mjs` qui contient un tableau de produits.

Création du fichier `mock-product.mjs` dans le répertoire `db`

```
let products = [
  {
    id: 1,
    name: "Big Mac",
    price: 5.99,
    created: new Date(),
  },
  ...
];

export { products }
```

Création du fichier `products.mjs` dans le répertoire `routes`

Dans ce fichier, on trouvera toutes les routes des produits commençant par '/api/products/'.

La 1ère route est la route permettant de récupérer tous les produits grâce à la requête HTTP :

GET http://localhost:3000/api/products/

```
import express from "express";

import { products } from "../db/mock-product.mjs";

import { success } from "./helper.mjs";

const productsRouter = express();

productsRouter.get("/", (req, res) => {
  const message = "La liste des produits a bien été récupérée.";
  res.json(success(message, products));
});

export { productsRouter };
```

La méthode `success()`

La méthode `success()` a pour but de retourner un objet js qui contient un message et les données au format json.

```
const success = (message, data) => {
  return {
    message: message,
    data: data,
  };
};
```

```
export { success };
```

Retourner du json

Lorsque le consommateur de notre API REST fait un GET http://localhost:3000/api/products/, on doit lui retourner le json des produits.

C'est pour cela que l'on utilise `res.json`

`res` pour response c'est à dire la Réponse HTTP. `json()` la méthode qui retourne l'objet js en json.

Content-Type

Il est important que la réponse HTTP soit bien du json et pas du texte qui ressemble à du json.

Pour cela, il faut s'assurer que la méthode json à bien fait son job.

Un des moyens les plus couramment utilisé est de faire un `curl`.

Lorsque vous pensez que votre code est terminé, faites la commande suivante :

```
curl -i http://localhost:3000/api/products/
```

```
Greg@LAPTOP-5335QTOF MINGW64 ~/OneDrive - Education Vaud/295/self-service-machine-api
$ curl -i http://localhost:3000/api/products
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   765  100   765    0     0  23360      0 --:--:-- --:--:-- --:--:-- 23906HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 765
ETag: W/"2fd-dMhWz5REr8y/c1FeiY7WTEOgsYI"
Date: Wed, 27 Dec 2023 08:56:33 GMT
Connection: keep-alive
Keep-Alive: timeout=5

{"message":"La liste des produits a bien été récupérée.","data":[{"id":1,"name":"Big Mac","price":5.99,"c
reated":"2023-12-27T08:56:17.080Z"},{"id":2,"name":"Mc Chicken","price":4.99,"created":"2023-12-27T08:56:
17.080Z"},{"id":3,"name":"Double Cheese Burger","price":2.99,"created":"2023-12-27T08:56:17.080Z"},{"id":
4,"name":"Fries","price":2.99,"created":"2023-12-27T08:56:17.080Z"},{"id":5,"name":"Mc Nuggets","price":3
.49,"created":"2023-12-27T08:56:17.080Z"},{"id":6,"name":"Salad","price":2.79,"created":"2023-12-27T08:56
:17.080Z"},{"id":7,"name":"Coke","price":1.99,"created":"2023-12-27T08:56:17.080Z"},{"id":8,"name":"Ice T
ea","price":1.99,"created":"2023-12-27T08:56:17.080Z"},{"id":9,"name":"Water","price":1.49,"created":"202
3-12-27T08:56:17.080Z"}]}
```

Comme vous pouvez le voir le `Content-Type` est bien `application/json`

Les points d'entrées / et /api/

Pour terminer, nous avons simplement fait une redirection de '/api/' vers '/'.

Voila le fichier `src/app.mjs` complet :

```
import express from "express";

const app = express();
const port = 3000;

app.get("/", (req, res) => {
  res.send("API REST of self service machine !");
});
```

```
app.get("/api/", (req, res) => {
  res.redirect(`http://localhost:${port}/`);
});

import { productsRouter } from "./routes/products.mjs";
app.use("/api/products", productsRouter);

app.listen(port, () => {
  console.log(`Example app listening on port http://localhost:${port}`);
});
```

Passons à l'étape n°3